УДК 004.8:004.9

М.В. Крыжановский, М.Ю. Мальсагов
НИИ системных исследований РАН, г. Москва, Россия
iont.niisi@gmail.com, Magomed.Malsagov@gmail.com

Особенности применения дискретизации в задачах поиска глобального минимума[1]

В работе рассматривается задача нейросетевой минимизации квадратичного функционала в пространстве бинарных переменных. Для решения задачи предложен и исследован модифицированный алгоритм минимизации, основанный на применении метода дискретизации. Показано, что его применение дает уменьшение объема вычислений.

Введение

В настоящей работе рассматривается задача минимизации квадратичного функционала

$$E = -(\mathbf{s}, \mathbf{A}\mathbf{s}) + 2 \cdot (\mathbf{B}, \mathbf{s}), \tag{1}$$

в котором компоненты вектора **s** принимают значения ±1. Матрица $\mathbf{A}$ – симметричная, с нулевой диагональю, а ее элементы – случайные независимые величины. Количество минимумов функционала $E(\mathbf{s})$ велико и экспоненциально растет с ростом размерности задачи. Так уже при размерности вектора **s** равной 200 число минимумов ~100000 и поэтому нахождение достаточно глубокого минимума сложная задача.

Алгоритм минимизации. За основу процедуры минимизации взята нейросетевая модель Хопфилда [1]. Это полносвязная рекуррентная нейронная сеть из N нейронов, имеющих два состояния $s_i = \pm 1$, $i = \overline{1, N}$. Энергия сети задана выражением (1). Такую сеть можно рассматривать как систему, решающую задачу бинарной минимизации: конвергируя в устойчивое состояние, сеть находит конфигурацию, соответствующую минимуму энергии E. Показано [2], [3], что при случайном поиске вероятность отыскания какого-либо минимума экспоненциально возрастает с ростом глубины этого минимума. Это означает, что нейросеть с подавляющей вероятностью находит если не оптимальное решение (глобальный минимум), то одно из субоптимальных решений (локальный минимум).

Мы будем рассматривать только асинхронную динамику сети Хопфилда, однозначно приводящую к минимизации функционала энергии E: на каждом такте работы сети вычисляется одна из компонент (например, i-я) локального поля $\mathbf{H}$

$$\mathbf{H} = -\mathbf{B} + \mathbf{A} \cdot \mathbf{s}, \tag{2}$$

и компоненте s_i присваивается значение

$$s_i = sign\, H_i. \tag{3}$$

Эта процедура последовательно применяется ко всем компонентам $\boldsymbol{s}$ до тех пор, пока сеть не конвергирует в устойчивое состояние, соответствующее минимуму энергии.

[1] Работа поддержана проектом 2.15 Президиума РАН и грантом РФФИ 09-07-00159.

Для уменьшения объема вычислений в работе [4] предложен метод дискретизации матрицы $\mathbf{A}$, т.е. выделение из матричных элементов среднего значения A_0 и замены остатка матрицей $\mathbf{C}$, нормированные элементы которой имеют целочисленные значения в диапазоне $[-m;+m]$, где m – число градаций. При таком подходе поиск минимума $E(\mathbf{s})$ заменяется поиском минимума дискретизованного функционала

$$\varepsilon = -(\mathbf{s}, \mathbf{Cs}) + 2 \cdot (\mathbf{b}, \mathbf{s}), \tag{4}$$

если $A_0 = 0$.

Оптимизация функционала (4) проводится аналогично оптимизации функционала (1). На каждом такте работы вычисляется одна из компонент локального поля $\mathbf{h}$

$$\mathbf{h} = -\mathbf{b} + \mathbf{C} \cdot \mathbf{s} \tag{2a}$$

и компоненте присваивается значение

$$s_i = sign\, h_i\,. \tag{3a}$$

Качество такой аппроксимации будет зависеть от выбранного числа градаций и определяется функцией распределения. С увеличением числа градаций максимум функции распределения смещается влево (рис. 1), в сторону более глубоких минимумов, что увеличивает вероятность их нахождения. Так, для «энергии» $\delta E = 0{,}05$ имеем: при переходе от $m = 1$ к $m = 8$ плотность состояний увеличится в 1,5 раз. На рис. 1 представлены функции распределения плотности вероятности по «энергии» функционала (1) построенных на минимумах $\varepsilon(\mathbf{s})$ для разного числа градаций. По оси X отложена «энергия», которая определяется величиной $\delta E = (E_0 - E)/E_0$, где E_0 – энергия глобального минимума.

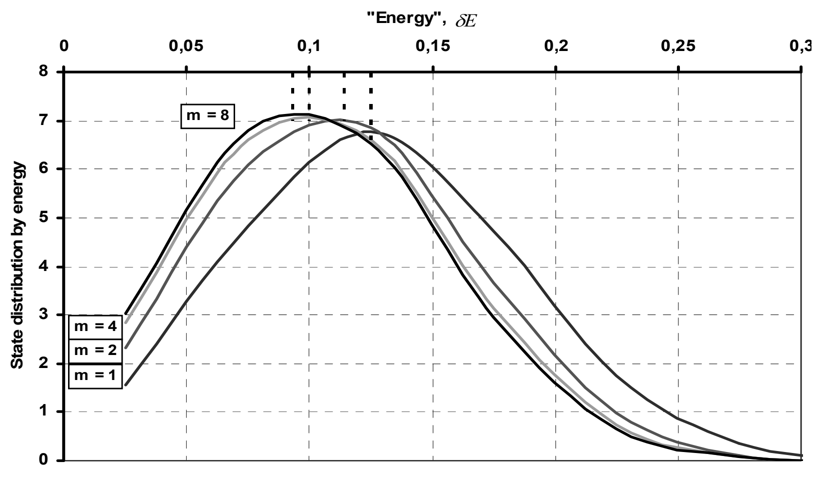


Рисунок 1 – Распределение плотности вероятности состояний по энергии для разного числа градаций

Минимум функционала. В данной работе применяется двухэтапный алгоритм [5]. Процесс минимизации функционала (1) начинается с некоторой случайной точки пространства $\mathbf{s}$. Подчиняясь решающему правилу (3а), сеть конвергирует в некоторое устойчивое состояние $\mathbf{s}_0^*$, являющееся минимумом функционала ε. Если из этой точки продолжить спуск с решающим правилом (3), то сеть конвергирует в состояние $\mathbf{s}_0$, соответствующее

минимуму функционала (1). На всем пути $\mathbf{s} \to \mathbf{s}_\mathbf{o}^* \to \mathbf{s}_\mathbf{o}$ величина ошибки (вероятность ошибки P в направлениях градиентов) только уменьшается. С ростом числа градаций m величина ошибки убывает как

$$P = \frac{1}{\pi(2m+1)}. \tag{4}$$

Эффективность минимизации определяется величиной $\delta E = \left(E_0^* - E_0\right)/E_0$, где $E_0^* = E(\mathbf{s}_\mathbf{o}^*)$ – энергия минимума $\mathbf{s}_\mathbf{o}^*$ дискретизированного функционала, а $E_0 = E(\mathbf{s}_\mathbf{o})$ – энергия ближайшего к нему минимума $\mathbf{s}_\mathbf{o}$, в который мы бы спустились при использовании исходного решающего правила (3). Показано [4], что эффективность минимизации

$$\delta E \approx P, \tag{5}$$

из которого следует, что разница в энергиях с ростом m убывает как $\delta E \sim m^{-1}$.

При этом Хеммингово расстояние между минимумами функционала E и его аналога ε определяются простым соотношением

$$d = NP. \tag{6}$$

Хеммингово расстояние не превышает значения $d = 0{,}11N$, когда $m = 1$, и снижается до значения $d = 0{,}02N$ при $m = 16$.

Сказанное подкрепляется результатами эксперимента (рис. 2). На рис. 2 кривые 1, 2, 4 – функции распределения состояний: начального $\mathbf{s}$, дискретизированного $\mathbf{s}_\mathbf{o}^*$ и конечного $\mathbf{s}_\mathbf{o}$. Кривая 3 – распределение состояний исходного функционала. На рис. 2 они же отмечают точки максимумов функций плотности. Расстояние между этими максимумами на графике соответствует в точности формуле (5). Видно, что величина первого участка пути $\mathbf{s} \to \mathbf{s}_\mathbf{o}^*$ значительно больше второго участка $\mathbf{s}_\mathbf{o}^* \to \mathbf{s}_\mathbf{o}$. Преимущество двухэтапного алгоритма в том, что первая часть пути $\mathbf{s} \to \mathbf{s}_\mathbf{o}^*$ проходится с большей скоростью. Вследствие чего достигается общее увеличение скорости работы алгоритма. Так же функция распределения минимумов 2-этапного алгоритма лежит левее распределения исходного алгоритма. Таким образом, вероятность нахождения глубоких минимумов двухэтапным алгоритмом выше, чем простым методом.

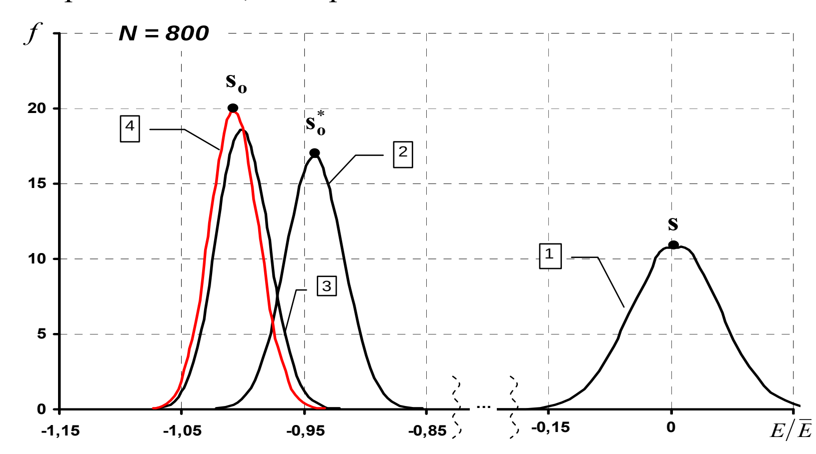


Рисунок 2 – Функции распределения состояний: начального $\mathbf{s}$ (кривая 1), дискретизированного $\mathbf{s}_\mathbf{o}^*$ (кривая 2) и конечного $\mathbf{s}_\mathbf{o}$ (кривая 4). Кривая 3 – распределение состояний исходного функционала

Работа с укороченной арифметикой

Полученные результаты дают основу для применения метода дискретизации, что означает возможность применения чисел малой разрядности. При этом для хранения матричного элемента при выборе параметра $m=1$ достаточно 2 битов, а для $m=15$ достаточно половины байта. В дальнейшем, размер чисел матрицы **A** мы будем называть исходным форматом, а p – количество чисел малой разрядности, которое упаковывается в исходный 4-байтный формат.

Основной объем вычислений по алгоритму Хопфилда приходится на вычисление градиента, т.е. на вычисления скалярных произведений 2 векторов. Поэтому возможность такого представления элементов **A** позволяет увеличить быстродействие алгоритма. Например, сложение пары вещественных чисел в 4-байтном формате требует 1 такт процессорного времени. За то же время можно сложить 4 пары целых чисел в байтном формате.

Ускорение алгоритма определяется величиной

$$\theta = \frac{I_0}{I},$$

где I_0 – время работы алгоритма, I – время работы алгоритма, когда элементы матрицы упакованы.

На рис. 3 приведен пример увеличения скорости работы алгоритма ($m=1$, параметр упаковки $p=8$). По оси ординат отложено ускорение θ, а по оси абсцисс – размерность нейросети N. С увеличением размерности нейросети его ускорение увеличивается и достигает своего предельного значения $\theta \approx 7{,}3$.

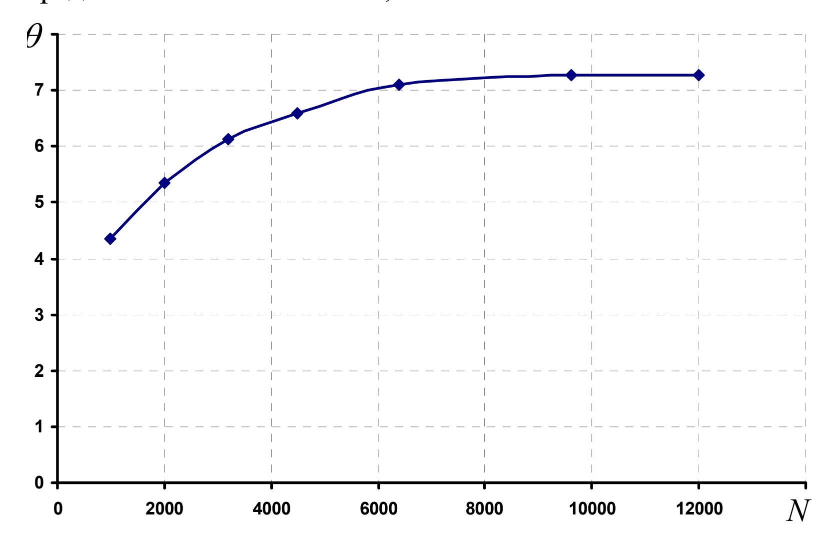


Рисунок 3 – Ускорение алгоритма при использовании упаковки чисел малой разрядности

На скорость работы сети так же влияет исходный формат чисел. Например, если элементы исходной матрицы **A** вещественные (4 байта), то время работы возрастет до 1.4, а для вещественных 10 байтных чисел возрастет до 6,3.

Алгоритм расчета нейронной сети Хопфилда основан на последовательном вычислении компонент градиента **H** при каждом изменении состояния нейрона.

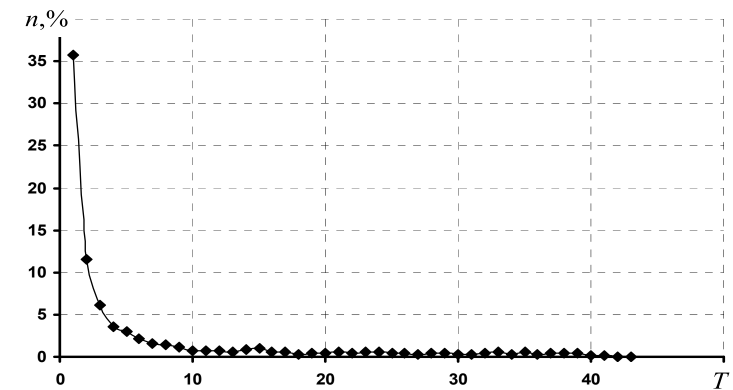


Рисунок 4 – Количество переворотов спинов на каждой итерации.
Размерность нейросети $N = 4000$

На рис. 4 показано, как изменяется число изменений состояний нейронов от номера итерации (одна итерация соответствует проверке всех N нейронов). На вертикальной оси отложена доля измененных нейронов n в процентах. Как видно из рис. 4, с увеличением числа шагов число нейронов, которые изменяют свое направление, становится малым. Это означает, что направления компонент $\mathbf{H}$ также редко изменяется. Поэтому вычисление на каждом шаге компоненты вектора $\mathbf{H}$ не эффективно. Вычислительная сложность алгоритма $\sim 2TN^2$, где T – число итераций. С увеличением размерности сети число итераций T растет (рис. 5b). Например, при размерности $N = 100$ число шагов $T = 5$, а при $N = 10000$ число шагов $T = 80$.

В настоящее время используется другой метод расчета. В исходном состоянии $\mathbf{s}$ вычисляются все компоненты $\mathbf{H}$. На каждом шаге процедуры при изменении состояния нейрона вектор $\mathbf{H}$ модифицируется по правилу $\mathbf{H} = \mathbf{H} \pm 2(\mathbf{A})_i$, если направление спина положительно/отрицательно. $(\mathbf{A})_i$ – вектор строки с номером текущего нейрона. Объем вычислений компонент $\mathbf{H}$ в исходном состоянии равен N^2. При перевороте каждого спина производится N операций. Число переворотов спинов не превышает N (рис. 5a), поэтому вычислительная сложность такого алгоритма $\sim (1+\alpha)N^2$, а $\alpha < 1$. Отношение объемов вычислений такого метода к исходному методу $\sim T, (T >> 1)$. В дальнейшем, в качестве исходного алгоритма мы используем метод с обновлением компонент $\mathbf{H}$.

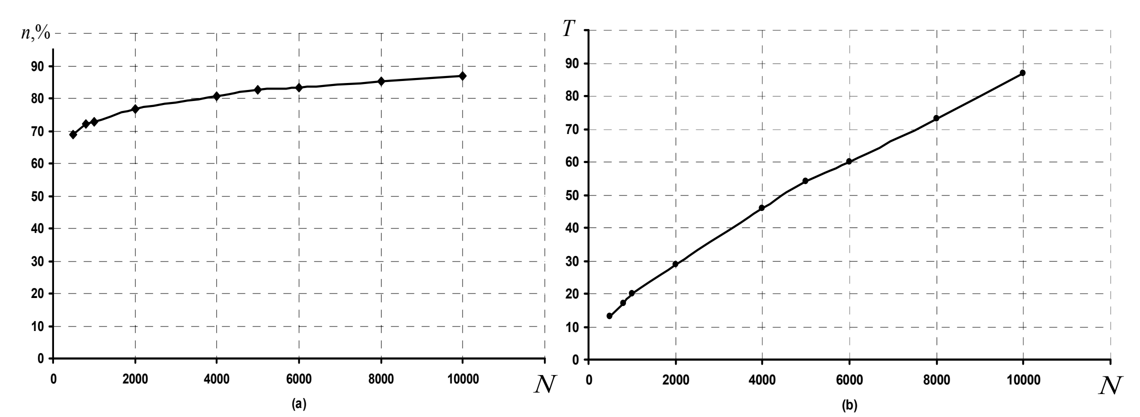


Рисунок 5 – (a) – Полное число изменений нейронов при увеличении размерности нейронной сети N; (b) – Изменение числа итераций в зависимости от размерности сети N

Модификация алгоритма обновления при использовании чисел малой разрядности достигается их упаковкой в исходный формат. В результате этого достигается параллельное выполнение операций сложения и умножения, что приводит к увеличению скорости работы. На рис. 6 показано ускорение алгоритма при упаковке $(p = 8,\ m = 1)$ для нейронных сетей различной размерности. Установлено, что достигаемое ускорение $\theta \approx 5$. При упаковке $p = 4$ в исходный формат ускорение составляет $\theta \approx 3{,}8$ $(m < 15)$, а при $p = 16$ $(m = 1)$ – $\theta \approx 8$.

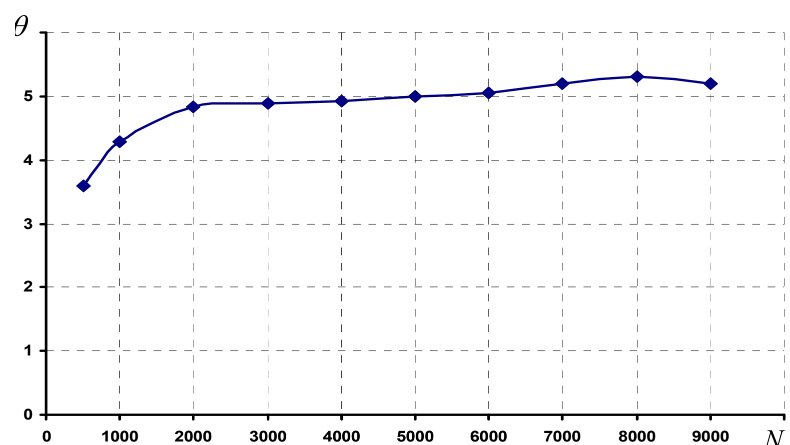


Рисунок 6 – Ускорение, получаемое при упаковке чисел $(p = 8)$ по сравнению с обычным методом

Алгоритм случайного поиска

Проведем анализ быстродействия двухэтапного алгоритма при использовании упакованных данных $(m = 1,\ p = 8)$ на первом этапе. В этом случае

$$\theta = \frac{I_0}{I} = \frac{O^H}{O_1 + O_2}.$$

На I этапе число итераций при минимизации $\varepsilon(\mathbf{s})$ равно числу итераций исходным методом [5]. Поэтому объем вычислений на первом этапе – $O_1 = O^H/5$. На II этапе число итераций T мало [5], соответственно мало число изменений в состояниях нейронов. Поэтому основной объем вычислений приходится на вычисление градиента $\mathbf{H}(\mathbf{s}_\mathbf{o}^*)$ в промежуточных состояниях и составляет $O_2 = O^H/2$. В этом случае ускорение алгоритма составит $\theta \approx 1{,}4$.

Лимитирующим фактором, препятствующим увеличению быстродействия 2-этапного алгоритма, является объем вычислений градиентов $\mathbf{H}$ промежуточных состояний $\mathbf{s}_\mathbf{o}^*$. Для его уменьшения на II этапе предлагается использование наиболее глубоких состояний $\mathbf{s}_\mathbf{o}^*$. Например, если мы продолжим спуск только с половины всех точек, то ускорение составит

$$\theta \approx \frac{O^H}{O_1 + 0{,}5 O_2} \approx 2{,}2.$$

Конечно, при таком подходе есть шанс получить несколько худшее решение. Вероятность этого можно оценить, используя экспериментальные результаты (рис. 7).

Для этого распределение минимумов $\mathbf{s}_0^*$ разбивалось на левую и правую области относительно пика распределения, и определялась вероятность найти самый глубокий минимум при стартах из каждой области. На рис. 7 видно, что самый глубокий минимум с вероятностью 96% находится из состояний левой области, в то время как правая область дает вероятность близкую к нулю. Поэтому состояния правой области могут не участвовать на II этапе поиска.

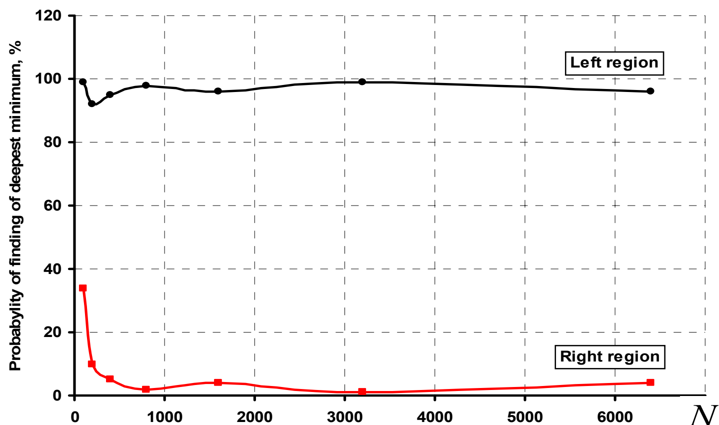


Рисунок 7 – Вероятность попадания в самый глубокий минимум из левой и правой областей распределения $\mathbf{s}_0^*$

Качество алгоритма зависит от размера области состояний $\mathbf{s}_0^*$, которые используются на II этапе, и определяется эффективным расстоянием $\delta E = \left(E_0^* - E\right)/ E_0^*$, где E_0^* – самый глубокий минимум, найденный со всех состояний $\mathbf{s}_0^*$, E – самый глубокий минимум, найденным при стартах из некоторой части состояний $\mathbf{s}_0^*$. Начало области соответствует наиболее глубокому минимуму дискретизированной сети. Конечная точка (граница области) – свободный параметр.

С ростом величины стартовой области эффективное расстояние монотонно убывает (рис. 8), а качество минимизации улучшается, и после некоторого значения обращается в ноль, т.е. в этой области находится самый глубокий минимум.

С помощью δE можно оценить Хемминогово расстояние D между глобальным минимумом E_0^* и самым глубоким минимумом E, найденным при стартах из отрезка. Действительно, при изменении состояния одного нейрона, энергия меняется в среднем на величину e. Поскольку большая часть минимумов находится в области $E_c \pm 3\sigma_E$ и отношение $\sigma_E / E_c << 1$, то $E_0^* \sim E_c$. $E_c \sim eN$. Тогда

$$\delta E \sim \frac{D}{N}.$$

Таким образом, достаточно подобрать область состояний $\mathbf{s}_0^*$, для которой выполняется условие

$$D = \delta E \cdot N \sim 1. \quad (7)$$

Для размерности сети $N = 100$ это условие будет выполняться в случае, когда граница области отстоит от максимума функции распределения на расстоянии $1{,}8\sigma_\varepsilon$. С увеличением размерности сети при фиксированном эффективном расстоянии размер области уменьшается (рис. 9), ее граница смещается в сторону более глубоких дискретизированных минимумов, что уменьшает объем вычислений.

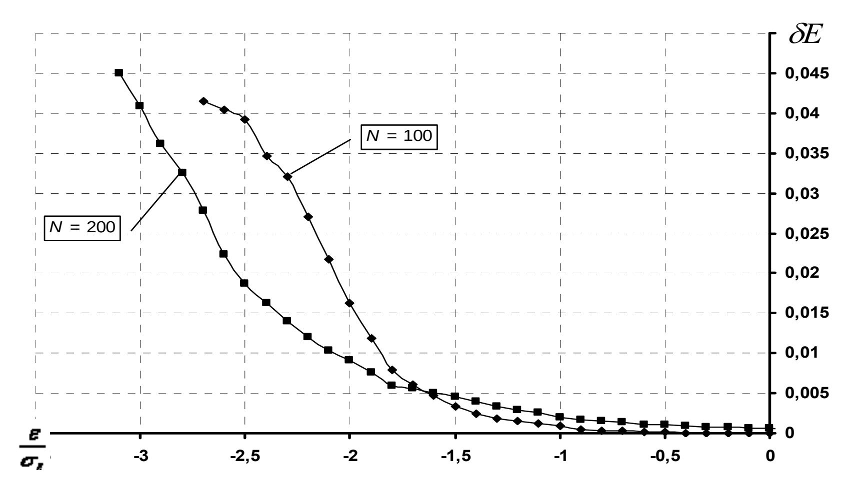


Рисунок 8 – Зависимость эффективного расстояния от размера стартовой области. Размер стартовой области задается в единицах σ_ε функции распределения энергии $\varepsilon\left(\mathbf{s}_\mathbf{o}^*\right)$

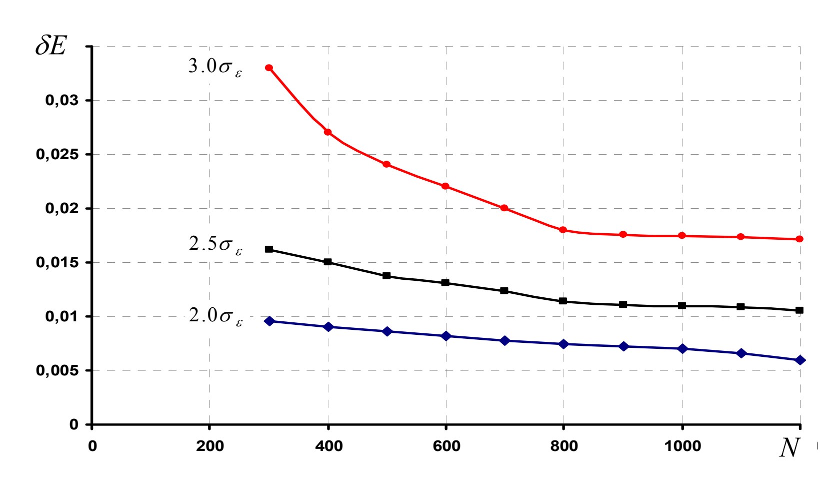


Рисунок 9 – Эффективное расстояние в зависимости от размерности сети для трех значений размера стартовой области

Действительно, распределение промежуточных состояний подчиняется нормальному закону. Количество минимумов $\mathbf{s}_\mathbf{o}^*$, находящихся на расстоянии более $2{,}0\sigma_\varepsilon$ от максимума функции распределения к общему их числу составляет меньше 0,05. В этом случае увеличение скорости работы алгоритма составит

$$\theta \approx \frac{O^H}{O_1 + 0{,}05 O_2} \approx 4{,}5 .$$

Данная величина ускорения верна для размерностей нейронной сети более 200.

Сравнение работы 2-этапного алгоритма с обычным (рис. 10) показывает, что модификация алгоритма увеличивает вероятность нахождение глубоких минимумов. Так, на рис. 10 представлены две кривые (экспериментальные результаты). Верхняя кривая соответствует вероятности нахождения 2-этапным алгоритмом лучшего или равного минимума, чем алгоритмом Хопфилда. Нижняя кривая – вероятность того, что найденный минимум хуже, чем при простом алгоритме. Видно, что с увеличением размерности сети вероятность найти более глубокий минимум модифицированным алгоритмом всегда больше 0,5 по сравнению с исходным алгоритмом.

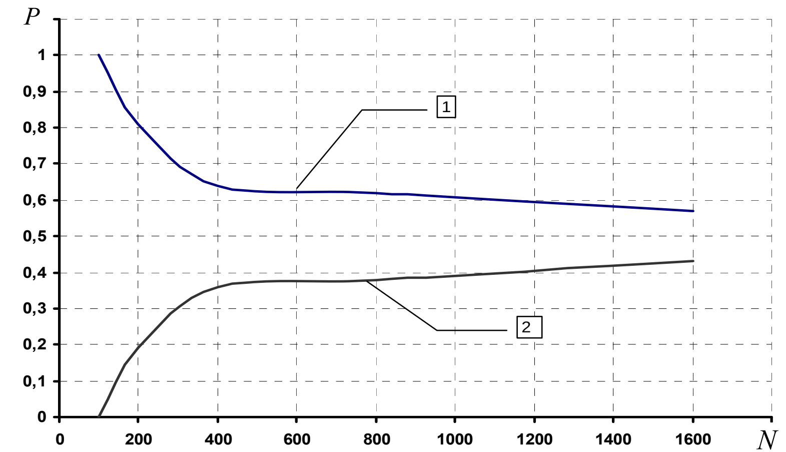


Рисунок 10 – Вероятность найти более (кривая 1) глубокий минимум или менее (кривая 2) глубокий с помощью 2-этапного алгоритма

Выводы

Предложена процедура дискретизации, позволяющая использовать числа малой разрядности для увеличения быстродействия процедуры минимизации квадратичного функционала. Для этого был разработан 2-этапный алгоритм на основе нейросетевой модели Хопфилда. Ускорение работы алгоритма для сетей размерностью более 200 превышает 4,5. Качество 2-этапного алгоритма по нахождению глубоких минимумов выше, чем у исходного метода.

Литература

1. Hopfield J.J. Neural networks and physical systems with emergent collective computational abilities / J.J. Hopfield // Proc. Nat. Acad. Sci. USA. –1982. – Vol. 79. – P. 2554-2558.
2. Крыжановский Б.В. Взаимосвязь глубины локального минимума и вероятности его нахождения в обобщенной модели Хопфилда / Б.В. Крыжановский, Б.М. Магомедов, А.Л. Микаэлян // ДАН. – 2005. – Т. 405, № 3. – С. 320-324.
3. Kryzhanovsky B.V. The shape of a local minimum and the probability of its detection in random search / B.V. Kryzhanovsky // Lecture Notes in Electrical Engineering. – 2009. – Vol. 24. – P. 51-61.
4. Крыжановский Б.В. Дискретизация матрицы в задаче бинарной минимизации квадратичного функционала / Б.В. Крыжановский, В.М. Крыжановский, М.Ю. Мальсагов // ДАН. – 2011. – Т. 438, № 3. – С. 312-317.
5. Kryzhanovsky M.V. Clipping procedure in optimization problems and its generalization / M.V. Kryzhanovsky, M.Yu. Malsagov // Optical Memory & Neural Networks. – 2009. – Vol. 18, № 3. – P. 181-187.

Literatura

1. Hopfield J.J. Proc. Nat. Acad. Sci. USA. 1982. V. 79. P 2554-2558.
2. Kryzhanovskij B.V. DAN. T. 405. № 3. 2005. S. 320-324.
3. Kryzhanovsky B.V. Lecture Notes in Electrical Engineering. V. 24. 2009. P. 51-61.
4. Kryzhanovskij B.V., DAN. T. 438. № 3. 2011. S. 312-317.
5. Kryzhanovsky M.V. Optical Memory & Neural Networks. V. 18. № 3. 2009. P. 181-187.

M.V. Kryzhanovsky, M.Yu. Malsagov

Application Features of Discretization in the Problem of Global Minimum Search

The problem of neural network minimization of quadratic functional in binary space is considered in this paper. To solve problem, modified minimization algorithm is proposed and researched. This algorithm is based on discretization procedure. It is shown that its applying decrease amount of calculation.

Статья поступила в редакцию 22.06.2011.